



자동화



자동화란 무엇인가?



프로그래밍의 본질

프로그래밍 한 줄을 실행하는 시점부터 자동화를 수행하고 있습니다.



보안 솔루션의 기반

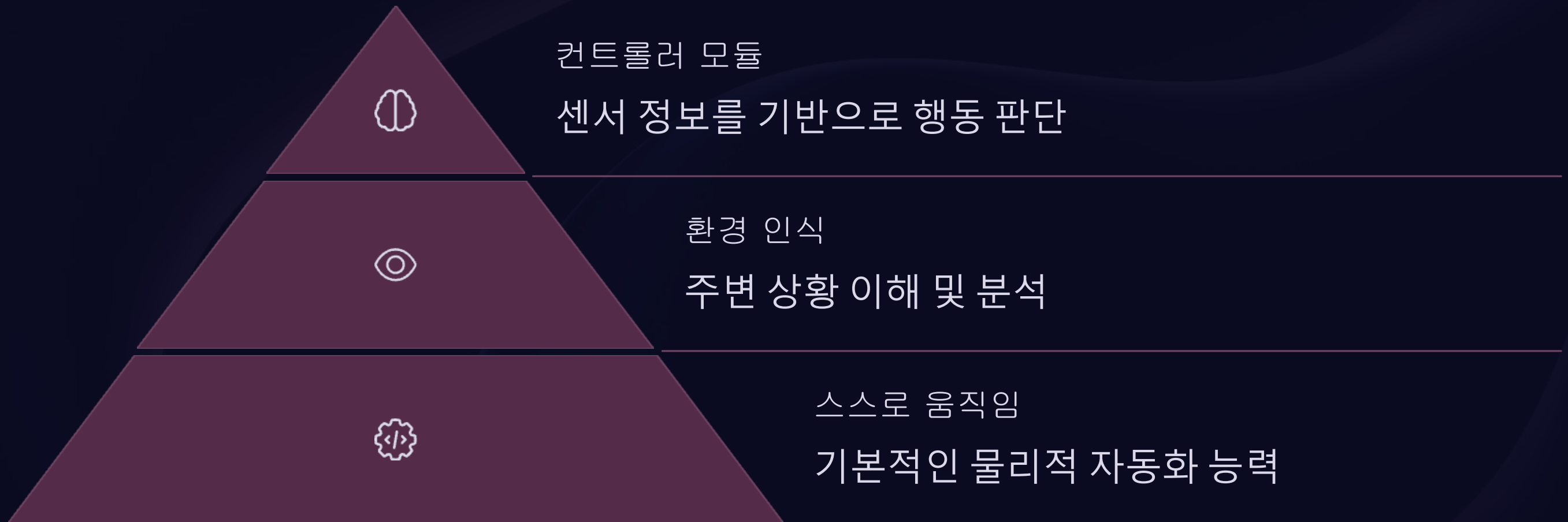
백신, 제어 솔루션, 모니터링 시스템, 방화벽 등은 보안적 지식을 자동화한 결과입니다.



효율성 추구

사람은 항상 효율성을 추구하기 때문에 많은 일들이 자동화를 향해 발전합니다.

자동화의 구성 요소



- 스스로 움직이는 능력
- 환경을 인식하는 센서,
- 판단하는 컨트롤러가 필요합니다.



자동화의 종류

1

화면 좌표 기반 자동화
화면 좌표 기반

2

UI 자동화 셀레늄, Pywinauto 등

3

모니터링 및 작업 자동화
모니터링 자동화

보안에서의 자동화

틈새 시장 발견

아직 변동성이 크거나
시장성이 없는 분야 탐색

성과 측정

자동화 효과 분석 및 개선



작은 도구 개발

일상의 작은 일들을 자동화
하는 도구 제작

조각 모으기

작은 조각들을 모아 큰 퍼즐 완성

개인이 보안 자동화에 접근할 때는 틈새 시장을 찾거나
일상의 작은 일들을 자동화하는 것부터 시작하는 것이 좋습니다.

셀레늄

구성	설명
webdriver	크롬, 엣지 같은 브라우저를 실제로 띄움
find_element()	HTML 요소를 찾음 (버튼, 입력창 등)
.click(), .send_keys()	클릭하거나 텍스트 입력

셀레늄은 웹 브라우저를 대신 움직여주는 로봇 손

셀레늄을 활용한 자동화

모듈 설치

pip install selenium 명령으로 셀레늄 모듈 설치

웹 드라이버 다운로드

크롬 브라우저 버전에 맞는 웹 드라이버를 다운로드/ 설치

코드 작성 및 실행

브라우저를 자동 제어하여 웹 페이지를 스캔 /결과 분석

셀레늄

- 웹 자동화 분야에서 많이 쓰이는 오픈 소스
- 브라우저를 직접 제어
- 방어를 회피하기 용이

SELENIUM WEB AUTOMATION

SELENIUM



셀레늄을 활용한 자동화

모듈 설치

pip install selenium 명령으로 셀레늄 모듈 설치

셀레늄을 활용한 자동화

웹 드라이버 다운로드

크롬 브라우저 버전에 맞는 웹 드라이버를 다운로드/설치

크롬 주소창에 `chrome://version/`

셀레늄을 활용한 자동화

웹 드라이버 다운로드

Chrome 버전:
136.0.7103.93




셀레늄을 활용한 자동화

웹 드라이버 다운로드

검색 :chrome web driver

chrome web driver


전체 이미지 동영상 쇼핑 짧은 동영상 뉴스 웹 : 더보기

 Chrome for Developers
https://developer.chrome.com > Docs > ChromeDriver

Downloads | ChromeDriver - Chrome for Developers

2024. 12. 17. — ChromeDriver 114.0.5735.20. Supports Chrome version 114. Resolved issue 4121: WebDriver command sometimes fails with "unexpected command ..."

Version selection Chrome Driver Getting Started Chrome Extensions Android

 Chrome for Developers
https://developer.chrome.com > Docs > ChromeDriver

Get started with ChromeDriver - Chrome for Developers

2025. 2. 10. — This page documents **how to start using ChromeDriver** for testing your website on desktop (Windows/Mac/Linux).|

ChromeDriver

ChromeDriver

개요

Capabilities 및 Chrome Options

Chrome 확장 프로그램

참여

디자인 문서

ChromeDriver 다운로드

Chrome 버전 115 이상

이전 버전의 Chrome

안정화 버전

카나리아 릴리스

버전 선택

translated by Google 이 페이지는 Cloud Translation API를 통해 번역되었습니다. [Switch to English](#)

홈 > Docs > ChromeDriver

도움이 되었나요?

다운로드

경고: Chrome 버전 115 이상을 사용하는 경우 테스트용 Chrome 사용 가능 여부 대시보드를 참고하세요. 이 대시보드는 특정 ChromeDriver 버전을 다운로드할 수 있는 JSON 엔드포인트가 있습니다.

이전 Chrome 버전

이전 버전의 Chrome을 지원하는 ChromeDriver 버전은 다음과 같습니다.

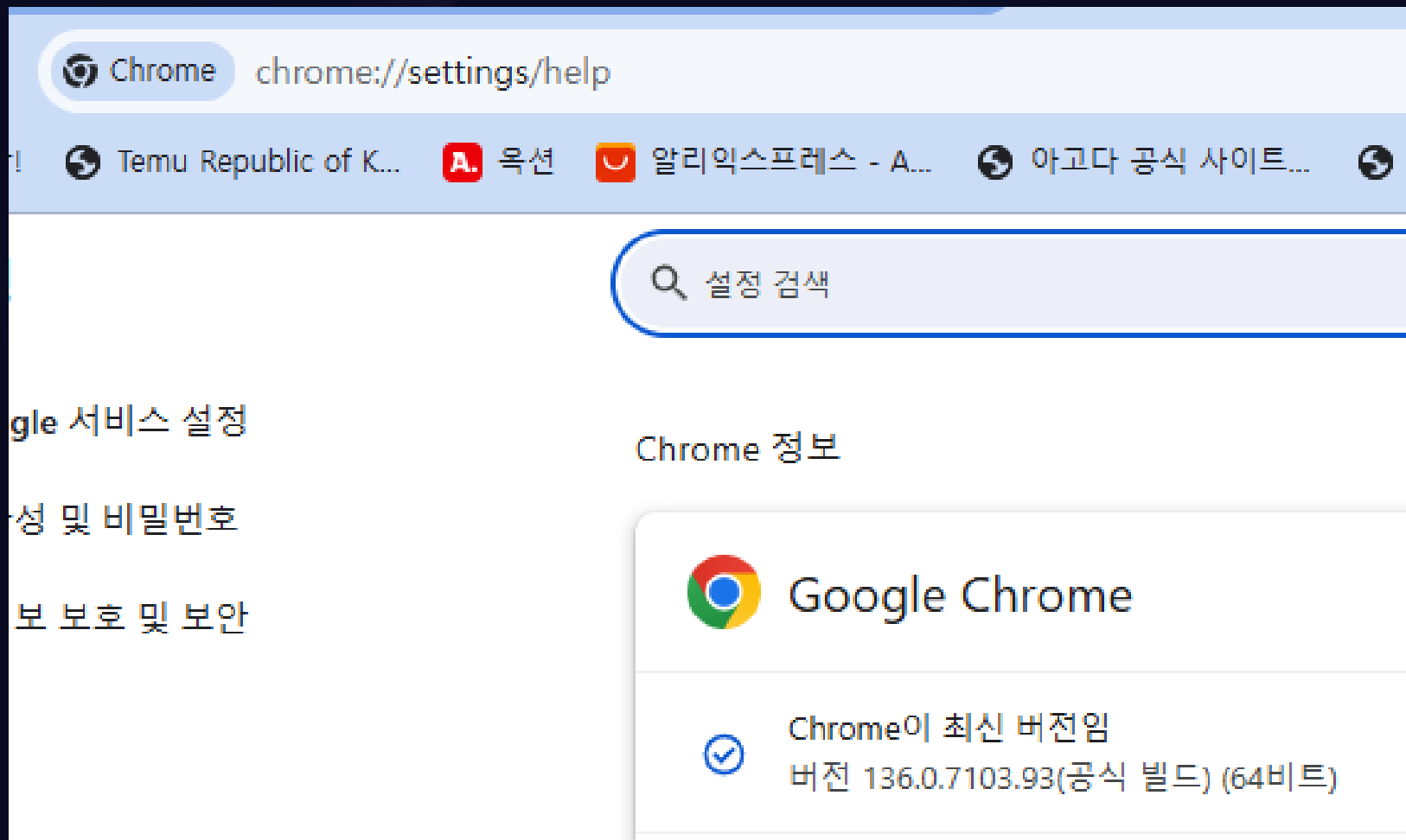
적절한 버전의 ChromeDriver를 선택하는 방법에 관한 자세한 내용은 [버전 선택](#) 페이지를 참고하세요.

ChromeDriver 114.0.5735.90

Chrome 버전 114 지원

셀레늄을 활용한 자동화

웹 드라이버 다운로드



Chrome 업데이트:
chrome://settings/help

셀레늄을 활용한 자동화

웹 드라이버 다운로드

Chrome for Testing availability

This page lists the latest available cross-platform Chrome for Testing versions and assets per Chrome release channel.

Consult [our JSON API endpoints](#) if you're looking to build automated scripts based on Chrome for Testing release data.

Last updated @ 2025-05-07T15:09:49.292Z

Channel	Version	Revision	Status
Stable	136.0.7103.92	r 1440670	✓
Beta	137.0.7151.6	r 1453031	✓
Dev	138.0.7153.0	r 1453769	✓
Canary	138.0.7166.0	r 1456678	✓

Stable

Version: 136.0.7103.92 (r1440670)

Binary	Platform	URL
chrome	linux64	https://storage.googleapis.com/chrome-for-testing-public/136.0.7103.92/linux64/chrome
chrome	mac-arm64	https://storage.googleapis.com/chrome-for-testing-public/136.0.7103.92/mac-arm64/chrome
chrome	mac-x64	https://storage.googleapis.com/chrome-for-testing-public/136.0.7103.92/mac-x64/chrome

chrome	linux64	https://storage.googleapis.com/chrome-for-testing-public/136.0.7103.92/linux64/chrome-linux64.zip
chrome	mac-arm64	https://storage.googleapis.com/chrome-for-testing-public/136.0.7103.92/mac-arm64/chrome-mac-arm64.zip
chrome	mac-x64	https://storage.googleapis.com/chrome-for-testing-public/136.0.7103.92/mac-x64/chrome-mac-x64.zip
chrome	win32	https://storage.googleapis.com/chrome-for-testing-public/136.0.7103.92/win32/chrome-win32.zip
chrome	win64	https://storage.googleapis.com/chrome-for-testing-public/136.0.7103.92/win64/chrome-win64.zip
chromedriver	linux64	https://storage.googleapis.com/chrome-for-testing-public/136.0.7103.92/linux64/chromedriver-linux64.zip
chromedriver	mac-arm64	https://storage.googleapis.com/chrome-for-testing-public/136.0.7103.92/mac-arm64/chromedriver-mac-arm64.zip
chromedriver	mac-x64	https://storage.googleapis.com/chrome-for-testing-public/136.0.7103.92/mac-x64/chromedriver-mac-x64.zip
chromedriver	win32	https://storage.googleapis.com/chrome-for-testing-public/136.0.7103.92/win32/chromedriver-win32.zip
chromedriver	win64	https://storage.googleapis.com/chrome-for-testing-public/136.0.7103.92/win64/chromedriver-win64.zip
chrome-headless-shell	linux64	https://storage.googleapis.com/chrome-for-testing-public/136.0.7103.92/linux64/chrome-headless-shell-linux64.zip

<https://storage.googleapis.com/chrome-for-testing-public/136.0.7103.92/win64/chromedriver-win64.zip>

Selenium 코드 작성

포털사이트 자동 접속1

Selenium 코드 작성
포털사이트 자동 접속2
자동요청 회피(구글)

Selenium 코드 작성
포털사이트 자동 접속3
자동요청 회피(네이버)

Selenium 코드 작성1

스캐너 코드 Selenium 코드 변경

START

브라우저 실행
(ChromeDriver)

타겟 페이지 열기:
"http://127.0.0.1:5000/"

<form> 태그 탐색
없음 → [종료]

form 태그 내 input 필드 수집

공격용 페이로드 목록 준비

[반복문 시작 →
각 페이로드마다 아래 수행]

form 페이지 다시 열기

input 필드에 페이로드 삽입

form 자동 전송 (submit)

에러 메시지(.errmsg) 존재 여부 확인

판단 기준 문자열 포함 여부 확인

있으면 "인젝션 발견" 출력

반복 종료 후 브라우저 닫기

END

Selenium 코드 작성2

pywinauto를 사용한 메모장 자동화

Selenium 코드 작성3

구글 주기적으로 검색예제

Selenium 코드 작성4

구글 주기적으로 검색 및 저장 예제

(pandas, excel 저장)

Selenium 코드 작성5
DB에서 개인정보 탈취
(pandas, excel 저장)

스캐너 (requests + BeautifulSoup 기반)

항목	설명
HTML 요청 방식	requests.get(url)로 HTML 원본만 가져옴 (브라우저 없이)
HTML 파싱	BeautifulSoup으로 <form>이나 <input> 태그 추출
폼 전송 방식	requests.post() 또는 requests.get()으로 수동 전송
동적 페이지 지원	JavaScript로 렌더링되는 요소는 못 가져옴
속도	매우 빠름 (브라우저 띄우지 않음)
용도	정적 페이지에서 자동 SQL 인젝션, 크롤링 등 빠른 테스트에 적합

스캐너 (Selenium 기반)

항목	설명
브라우저 작동	실제 브라우저를 띄우고 사람처럼 동작 (ChromeDriver 등)
HTML 파싱	find_element(By.XXX, ...) 등으로 요소 탐색
폼 전송 방식	.send_keys()로 입력 후 .submit() 또는 .click()
동적 페이지 지원	JavaScript로 로딩되는 SPA 페이지까지 지원
속도	상대적으로 느림 (브라우저 구동 포함)
용도	동적 페이지 테스트, UI 자동화, 로딩 지연 대응 등 정교한 조작에 적합

실전 비교 예시

구분	requests + bs4 스캐너	Selenium 스캐너
대상 페이지	정적 페이지에 적합	정적 + 동적 페이지 모두
QL Injection 탐지	가능	가능
로그인 후 폼 공격	세션 관리 복잡함	가능 (브라우저가 세션 유지함)
CAPTCHA 대응	불가능	자동화 탐지 시 우회 어려움
실제 사용자 시뮬레이션	X	마우스/키보드 이벤트까지 가능
속도	빠름	느림 (GUI 구동 포함)

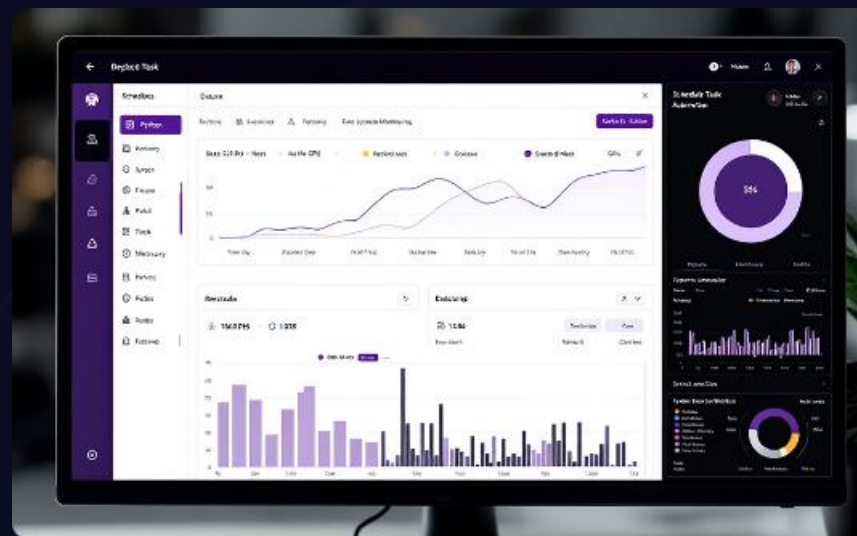
실전 예제들

requests + bs4 스캐너	Selenium 스캐너
공공 API 또는 HTML 기반 공공정보 페이지 크롤링	네이버 로그인 후 포털 검색어 자동 등록
블로그 제목, 본문 수집 (네이버, 브런치 등)	회사 ERP에서 업무보고 자동 등록
상품 가격 비교 (쇼핑몰에서 가격만 추출)	웹사이트 화면 자동 스크린샷 및 PDF 저장
	QA팀의 UI 테스트 자동화 (버튼 클릭, 결과 확인)
빠른 테스트, 단순한 입력 필드 공격	로그인 후 공격, 동적 폼/JS 필드 공격

시스템 UI 자동화



Pywinauto 활용
윈도우 애플리케이션의
객체 자동 제어



주기적 작업 실행
Advanced Python Scheduler를 사용
해 스케줄러 자동실행



데이터베이스 스캐닝
정규 표현식을 활용
데이터베이스 내 개인정보
자동 검색.